(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 January 2001 (11.01.2001)

PCT

(10) International Publication Number
WO 01/03400 A2

(51) International Patent Classification⁷: H04L 29/00

(21) International Application Number: PCT/IB00/00888

(22) International Filing Date: 1 July 2000 (01.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/346,747 2 July 1999 (02.07.1999) US

(71) Applicant *(for all designated States except US)*: **NOKIA INTERNET COMMUNICATIONS INC.** [US/US]; Robert C. Rolnik, 313 Fairchild Drive, Mountain View, CA 94043 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **RAISANEN, Vilho, I.** [FI/FI]; Laajavudrenrinne 6 E 29, FIN-01620 Vantaa (FI). **EKBERG, Jan-Erik** [FI/FI]; Seljatie 1 A 5, FIN-00320 Helsinki (FI).

(74) Agent: **ROLNIK, Robert**; Nokia Inc., 6000 Connection Drive, 1-4-755, Irving, TX 75039 (US).

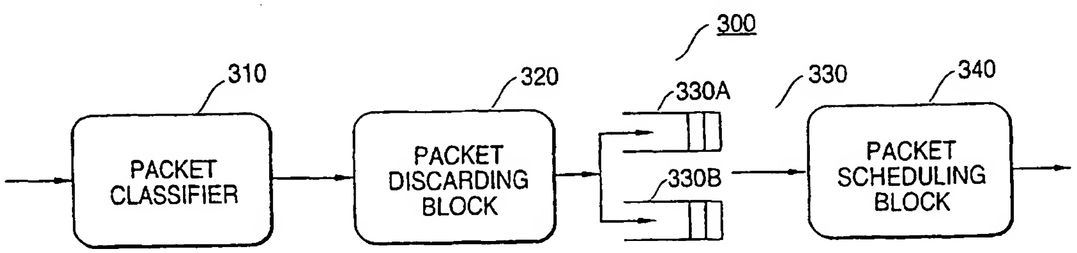
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: REAL-TIME TRAFFIC SHAPER WITH KEEP-ALIVE PROPERTY FOR BEST-EFFORT TRAFFIC



(57) **Abstract:** An advanced traffic shaper is provided for shaping real-time traffic in an IP-based network while simultaneously providing keep-alive bandwidth for best-effort traffic. The traffic shaper comprises a packet classifier coupled to receive an incoming packet from the network and to classify the incoming packet as one of a real-time packet and a best-effort packet; a packet discarding block coupled to perform traffic shaping of the incoming packet from the packet classifier and discard the incoming packet in accordance with traffic shaping parameters; connection queues coupled to temporary store real-time packets and best-effort packets from the packet discarding block, respectively; and a packet scheduling block coupled to send the real-time and best-effort packets from the connection queues to an outgoing device in accordance with the traffic shaping parameters.

REAL-TIME TRAFFIC SHAPER WITH KEEP-ALIVE PROPERTY FOR BEST-EFFORT TRAFFIC

BACKGROUND OF THE INVENTION

Technical Field

The present invention relates to traffic control in a data communication system and more particularly, relates to a traffic shaper for controlling traffic bandwidth usage in an IP (Internet Protocol) network to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications such as interactive audio and/or video while concomitantly providing keep-alive bandwidth for best-effort traffic.

Related Art

In recent years, increased usage of the Internet has resulted in scarcity of network capacity, and compromised performance of traditional applications. At the same time, new applications such as interactive audio and/or video, including video-conferencing or Voice over IP (VoIP) have emerged which demand much improved QoS (Quality of Service). As a result, network service providers (both enterprise and traditional ISPs) are pressured to provide different customers alternative levels of service.

One of the most promising of these services is Differentiated Services (DiffServ), which provides for classification of different flows of IP data packets and prioritizing scheduling of these IP flows in routers. These schemes work best when traffic bandwidth of individual end users is bounded and contains as small fluctuations as possible. The boundedness property of IP data packets can be achieved by using *traffic shaping*, or limiting the traffic bandwidth to an agreed-upon number of IP data packets per time unit and discarding excess traffic.

Several traffic shaping techniques may be provided to shape the traffic so that the traffic does not exceed a predetermined transmission rate for each destination in an IP network. Existing IP traffic shapers are designed to limit bandwidth usage of TCP (Transfer Control Protocol) traffic. Subsequently, those traffic shapers are not suitable for special requirements of high-priority, interactive real-time traffic with small packet size, such as video-conferencing or Voice over IP (VoIP). At the same time, it is desirable to prevent TCP connections from being exhausted of bandwidth. Accordingly, there is a need for an advanced traffic shaper advantageously configured for priority forwarding of real-time traffic while providing keep-alive bandwidth for "best-effort" traffic at the same time. Also needed is an advanced traffic shaper used together with QoS (Quality of Service) core network technology, such as an ATM network.

SUMMARY OF THE INVENTION

Accordingly, various embodiments of the present invention are directed to an advanced traffic shaper for shaping real-time traffic in an IP-based network while simultaneously providing keep-alive bandwidth for best-effort traffic. Such a traffic shaper may be an algorithm integrated into existing operating system (OS) of a host processor and/or a computer readable medium for use in a host processor for priority forwarding for real-time traffic and avoiding correlated loss for real-time streams while providing keep-alive bandwidth for best-effort traffic. In addition, the traffic shaper may be installed in the host processor of a source terminal or a destination terminal in an IP-based network which uses a bandwidth-limited IP link for providing connection between the source terminal and the destination terminal.

P4

In accordance with the present invention, the traffic shaper may comprise a packet classifier coupled to receive an incoming packet from the network and to classify the incoming packet as one of a real-time packet and a best-effort packet; a packet discarding block coupled to perform traffic shaping of the incoming packet from the packet classifier and discard the incoming packet in accordance with traffic shaping parameters; connection queues coupled to temporary store real-time packets and best-effort packets from the packet discarding block, respectively; and a packet scheduling block coupled to send the real-time and best-effort packets from the connection queues to an outgoing device in accordance with the traffic shaping parameters.

The connection queues may include a real-time queue for storing the real-time packets and a best-effort queue for storing the best-effort packets. Both real-time and best-effort queues correspond to packet buffers constructed from first-in-first-out (FIFO) memory for storing real-time packets and best-effort packets, respectively.

The packet discarding block may be executed to discard an incoming packet by steps of: determining whether the real-time and best-effort queues are empty; when the real-time and best-effort queues are empty, resetting internal counters and a scheduled transmission time of the next incoming packet; determining whether a real-time latency is greater than a maximum latency value of the real-time queue; when the real-time latency is greater than the maximum latency value of the real-time queue, discarding the real-time packet input from the packet classifier; determining whether a total latency is greater than a maximum latency value of both the real-time and best-effort queues; when the total latency is greater than the maximum latency value of both the real-time and best-effort queues, adding a best-effort packet to a tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue; determining whether a length of the real-time queue is greater than a maximum queue length of the real-time queue; when the length of the real-time queue is greater than a maximum queue length of the real-time queue, discarding the real-time packet input from the packet classifier; determining whether a length of the best-effort queue is greater than a maximum queue length of the best-effort queue; when the length of the best-effort queue is greater than a maximum queue length of the best-effort queue, adding a best-effort packet input from the packet classifier to the tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue; determining whether there is concurrency and time to discard a real-time packet to make room for best-effort packets; when there is concurrency and time to discard the real-time packet, discarding the real-time packet to implement the keep-alive bandwidth for best-effort packets; and placing the real-time packet that was not discarded in the real-time queue.

The packet scheduling block may be periodically executed to schedule output of an incoming packet by the steps of: determining whether both the real-time and best-effort queues are empty; determining whether there is concurrency and time to transmit the best-effort packet from the best-effort queue; when there is concurrency and time to transmit the best-effort packet, checking whether total bandwidth constraint allows transmission of the best-effort packet from the best-effort queue head; when transmission is allowed, updating a time of next transmission and transmitting the best-effort packet to the outgoing device; determining whether the real-time queue is not empty and time for transmission of the real-time packet is permitted; when the real-time queue is not empty and time for transmission of the real-time packet is permitted, updating the transmission time and transmitting the real-time packet to the outgoing device from the real-time queue head; determining whether the best-effort queue is not empty and time for transmission of the best-effort packet is permitted; and when the best-effort queue is not empty and time for transmission of the best-effort packet is permitted, updating the transmission time and transmitting the best-effort packet to the outgoing device from the best-effort queue head.

The present invention is more specifically described in the following paragraphs by reference to the drawings attached only by way of example.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the present invention, and many of the attendant advantages thereof, will become readily apparent as the same becomes better understood by reference to the following detailed description when considered in conjunction with the accompanying drawings in which like reference symbols indicate the same or similar components, wherein:

FIG. 1 illustrates an IP packet for use in an IP-based network according to the principles of the present invention;

FIG. 2 illustrates an IP-based network of a source terminal and a destination terminal having an advanced traffic shaper incorporated therein for controlling traffic bandwidth usage to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention;

FIG. 3 is a functional implementation of an advanced traffic shaper for controlling bandwidth usage to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention;

FIG. 4 is a flowchart representation of an IP packet discarding algorithm of the packet discarding block according to the principles of the present invention;

FIG. 5 is a flowchart representation of an IP packet scheduling algorithm of the packet scheduling block according to the principles of the present invention;

FIG. 6 illustrates the shaper output bandwidth of the advanced traffic shaper as a function of time;

FIG. 7 illustrates the length of real-time queue as a function of packet sequence number;

FIG. 8 illustrates the cummulative number of discarded real-time packets as a function of packet sequence number;

FIG. 9 illustrates one preferred usage implementation of an IP-based network of a source terminal and a destination terminal having an advanced traffic shaper incorporated therein for controlling traffic bandwidth usage to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention; and

FIG. 10 illustrates another preferred usage implementation of an IP-based network of a source terminal and a destination terminal having an advanced traffic shaper incorporated therein for controlling traffic bandwidth usage to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention.

DETAIL DESCRIPTION OF EMBODIMENTS OF THE INVENTION

The present invention is applicable for use with all types of data communication networks and communication services. However, for the sake of simplicity, discussions will concentrate mainly on an advanced traffic shaper for use in DiffServ-enhanced IP networks for providing guaranteed QoS (Quality of Service) for interactive real-time traffic of Voice over IP (VoIP), although the scope and implementation of the present invention is not limited thereto. For example, the novel traffic shaper may be implemented for real-time traffic of video or a combination of video and audio, such as video-conferencing for delivery of video over IP (although required bandwidth is larger but the basic problems associated with delivery of video over IP remain essentially the same as Voice over IP).

In addition, there may be several special network requirements for high-priority Voice over IP (VoIP) applications. For example, the most important requirement imposed by Voice over IP (VoIP) on the network delivery is limit on end-to-end delay. The exact upper limit varies depending on the source terminal, but 250 ms is an oft-quoted value for one-way delay limit. The end-to-end delay limit is not dependent on the chosen audio decoding scheme, although an audio codec may give rise to extra delay in the form of processing and framing.

Another requirement is a limit for packet loss. Typically, a single VoIP packet varies one or more voice samples (frames), and the loss of such a packet translates to several tens of milliseconds (ms) of lost audio data. The sensitivity on packet loss rate depends on the audio codec (encoder/decoder) used. The negative effects of packet loss can be reduced to some extent with means such as interpolating over a single missing sample, including Forward Error Correction (FEC) data - for example, a low-bit rate copy of a previous audio data frame -- into data packets. Regardless of the audio codec (encoder/decoder) and the interpolating means chosen in combating against packet loss, the audio quality may suffer or may be compromised after a sufficiently long sequence of missing packets. Hence if packet loss must be inflicted upon a VoIP stream, it is desirable to spread packets to be removed over time and to avoid correlated losses.

In addition to being able to prioritize real-time traffic, there are other desirable properties for the novel real-time traffic shaper. For usage with low-bandwidth links with mixed traffic types, it is useful to prevent best effort streams from being exhausted from bandwidth in case of real time stream nominal bandwidth being equal or larger than the link bandwidth. This way the time-outs associated with protocols such as FTP (File Transfer Protocol) can be avoided.

Attention now is directed to the drawings and particularly to FIG. 1, an IP packet for use in an IP-based network according to the principles of the present invention is illustrated. As shown in FIG. 1, an IP packet 2 consists of a segment of data 2B and a small IP header 2A prepended to the data 2B. The IP packet 2 may represent, but may not be limited to, real-time traffic of Voice over IP (VoIP), or best-effort (BE) traffic, such as FTP. The IP header contains, for example, IP addresses fields (32-bit global Internet address, generally consisting of a network identifier and a host identifier), a version field used to specify which version of the IP is represented in the IP packet (for example, IP Version 4 and IP Version 6), a type of service field used to specify how the IP packet is to be handled in IP-based networks which offer various service qualities, and a header checksum field used to verify transmission error. Other IP fields such as flags and fragment offset fields, a total length field, an ID field, a time to live field and a protocol field may also be included in such an IP header.

FIG. 2 illustrates an IP-based network of a source terminal 20 and a destination terminal 20 having an advanced traffic shaper incorporated therein for controlling traffic bandwidth usage to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention. As shown in FIG. 2, an IP bandwidth-limited link 10 is utilized to connect a source terminal 20 to a destination terminal 30. The IP bandwidth-limited link 10 may represent a low bandwidth link such as, for example, an integrated service digital service (ISDN) link and a modem link that connect to the source terminal 20 and the destination terminal 30. The IP bandwidth-limited link 10 may be connected at one end of an IP-based network of an Internet Service Provider (ISP), an Internet and different source and destination networks, including, for example, a public switched telephone network (PSTN) and an integrated services digital network (ISDN) that provide linkage between the ISP network ingress host and their receivers (see FIGs. 9 and 10). The source terminal 20 includes at least a host 22 and a network interface controller (NIC) 24. Likewise, the destination terminal 30 contains at least a host 32 and a network interface controller (NIC) 34.

In the source terminal 20, the host 22 generates data which is forwarded to the network interface controller (NIC) 24. The NIC 24 of the source terminal 20 transforms incoming data from host 22 into IP packets 2 and injects the IP packets 2 into the IP-based network 10. The rate of that IP packets are injected into the IP-based network 10 and the outward flow of IP packets 2 are controlled and managed by the NIC 24 to avoid congestion and to limit bandwidth usage of IP packets in the IP-based network 10. The IP-based network 10 accepts incoming IP packets 2 and forwards the same to destination terminal 30 according to the information contained in the IP header. The NIC 34 of the destination terminal 30 receives the IP packets 2 from the IP-based network 10, transforms the same into data and forwards the data to the host 32.

Included in the NIC 24 of source terminal 20 (also in the NIC 34 of destination terminal 20 for bidirectional communications on an end-to-end basis) is a novel traffic shaper implemented to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention. As contemplated by the present invention, the advanced traffic shaper is implemented by an algorithm configured from the beginning with a view on providing priority forwarding for real-time traffic and avoiding correlated loss for real-time streams while providing keep-alive bandwidth for best-effort traffic at the same time. The algorithm implementation may rely on the use of standard programming methods using high level program languages such as C or C++ and a general purpose (host) processor. The algorithm may be integrated into existing operating system (OS) of a host processor and/or a computer readable medium for use in a host processor for priority forwarding for real-time traffic and avoiding correlated loss for real-time streams while providing keep-alive bandwidth for best-effort traffic.

Referring now to FIG. 3, a functional block diagram of an advanced traffic shaper 300 constructed according to the principles of the present invention is illustrated. As shown in FIG. 3, the traffic shaper 300 includes several components, including a packet classifier 310, a packet discarding block 320, a set of connection queues 330 of real-time (RT) queue and best-effort (BE) queue for real-time (RT) packets and best-effort (BE) packets, and a packet scheduling block 340. These components may be embodied as software components or modules configured and/or written in accordance with the traffic shaping algorithm using those high level program languages such as C or C++ for priority forwarding for real-time traffic and avoiding correlated loss for real-time streams while providing keep-alive bandwidth for best-effort traffic in an IP-based network.

The packet classifier 310 is used to receive incoming IP packets 2 and to classify such incoming IP packets 2 as either real time (RT) packets or best effort (BE) packets. Classifier functionality is well known in IP architecture and therefore, needs not be described in detail herein.

The packet discarding block 320 is used to perform traffic shaping of an incoming packet from the packet classifier 310 and discard the incoming IP packet 2 if the packet fails to comply with bandwidth parameters (that is, too many input packets with respect to the shaper output bandwidth limited). The packet discarding block 320 is triggered to perform traffic shaping on an incoming packet 2 when an incoming packet 2 arrives from the packet classifier 310. The packet discarding block 320 contains packet counters used to balance real-time (RT) packets discards in order to achieve bandwidth for best-effort (BE) traffic in situations where there is concurrency.

The connection queues 330 includes a real-time (RT) queue 330A and a best-effort (BE) queue 330B for temporary storing real-time (RT) packets and best-effort (BE) packets respectively. In a preferred embodiment of the present invention, the connection queues 330 are packet buffers constructed from FIFO (first-in-first-out) memory for storing RT packets and BE packets, respectively.

The packet scheduling block 340 is used to send RT and/or BE packets to an outgoing device (e.g., host 34 of destination terminal 30) in accordance with bandwidth parameters.

The functionality of both the packet discarding block 320 and the packet scheduling block 340 of the advanced traffic shaper 300 is controlled by the following quantities:

- The length (l_p) and type (t_p) of input packet.
- The current point in time, denoted t.
- The queue lengths, len_{rt} and len_{be} for the real-time (RT), and best effort (BE) queues respectively.
- The shaper bandwidth, BW.
- Maximum latency values for the RT queue, and the complete queue system, denoted LAT_{rt} and LAT_{tot}, respectively.
- Maximum queue lengths for both RT and BE queues, denoted $QLEN_{rt}$ and $QLEN_{be}$. These value should be set large enough so that queue latency, not the queue length, is the cause for packet discarding.
- A time parameter that allows for compensation of low jitter with less exact shaping or vice versa. It is named t_{free}, and is explained in greater detail below in the context of algorithm description.
- Bandwidth for best-effort traffic which is maintained even at times when there is enough real time traffic to fill the whole pipe. Denoted $\mathrm{BW}_{\mathrm{BE,min}}$

In addition, a few internal variables are of interest for the functionality of the advanced traffic shaper 300. p_1 and p_2 are packet counters that are used to balance real-time packet discards (in order to achieve $\mathrm{BW}_{\mathrm{BE},min}$ for best-effort traffic) in situations where there is concurrency. t_{free} will indicate the next point in time when the advanced traffic shaper 300 is free to send the next packet according to the bandwidth specified.

The *pity* - parameters work as follows. In a state of concurrency (both RT and BE queues 330A-330B are active) each RT packet that passes through the traffic shaper 300 will add to the RT queue 330A according to the following formula:

$$p_1 \quad := \quad p_1 + (l_p * \mathrm{BW}_{\mathrm{BE,min}})/\mathrm{BW}$$

$$p_2 \quad := \quad p_2 + (l_p * \mathrm{BW}_{\mathrm{BE,min}})/\mathrm{BW}$$

which indicates the bit count of a transmission of equal time length as l_p transmitted over the traffic shaper bandwidth BW but calculated in $BW_{BE,min}$. In other words, a virtual channel is maintained for best-effort traffic, even though the bit count contributed by a single RT-packet is too small to fit a single BE packet.

Whenever a new RT packet arrives and the traffic shaper 300 is in a state of concurrency, the length of the packet is compared to p_1, and is discarded if the packet length is larger than p_1, and the p_1 will be decremented by the same amount. If a small flow of BE traffic is desired to be maintained even as the traffic shaper 300 is completely filled with RT traffic, some extra RT packets must be discarded to make room for the BE packets. In this manner, the RT packets that will be discarded will be spread out in the throughput.

The packet discarding block 320 is triggered on an incoming packet, and operates based on an IP packet discarding algorithm as follows:

```
Update the memory of the shaping algorithm if necessary
if $len_{rt}$ + $len_{be}$ = 0
   $\wedge(t - t_{free}) > t_{free}$
   then
        $t_{free}$=t - $t_{free}$;
        $p_1$= $p_2$ =0;
fi
Discard RT packets based on latency
if $latency(len_{rt})>LAT_{tot}$
        then
                discard (p);
fi;
Discard BE packets based on latency
if $latency(len_{rt}+len_{be})>LAT_{tot}$
        then
                add_to_best_effort (p);
                p := first_of_best_effort();
                discard (p);
fi;
Discard packets because queue is full
if$(t_p = REAL\ TIME) \wedge (len_{rt} \geq QLEN_{rt})$
        then
                discard (p);
```

```
fi;
if($t_p = BEST\ EFFORT) \wedge (len_{be} \geq QLEN_{be})$
        then
                add_to_best_effort ($p$);
                $p$ := first_of_best_effort();
                discard ($p$)
fi;
Discard real-time packets in a concurrency situation
if($t_p = REAL\ TIME) \wedge (l_p \leq p_1)$
        then
                $p_1 := p_1 - l_p$
                discard (p);
fi
add_to_queue (p);
```

Basically, the IP packet discarding algorithm as described provides several underlying functions: The first function relates to determining whether the real-time (RT) and best-effort (BE) queues 330A-330B are empty. If the RT and BE queues 330A-330B are empty, resetting counters p1 and p2 and the scheduled transmission time of the next packet. The second function relates to the RT packet and determining whether RT latency is too large. If the RT latency is too large, discarding the RT packet. The third function relates to the BE packet and determining whether the total latency is too large. If the total latency is too large, adding a BE packet to the tail of the BE queue 330B and discarding a packet from the BE queue head (best effort early warning property). The fourth function relates to the RT packet and determining whether the RT queue length is too large. If the RT queue length is too large, discarding the RT packet. The fifth function relates to the BE packet and determining whether the BE queue length is too large. If the BE queue length is too large, adding a BE packet to the tail of the BE queue 330B and discarding a BE packet from the BE queue head (early warning). The sixth function relates to the RT packet and determining whether there is concurrency (both RT and BE queues 330A-330B are active) and p1 counter indicates that it is time to discard a RT packet to make room for BE packets. If there is concurrency and p1 counter indicates that it is time to discard the RT packet, discarding the RT packet to implement the keep-alive bandwidth BW for BE packets. The seventh function relates to determining whether an incoming packet0 that was not discarded is placed in the appropriate queues 330A-330B.

Turning now to FIG. 4, a detailed flowchart representation of an IP packet discarding algorithm of the packet discarding block 320 is illustrated. The algorithm of the packet discarding block 320 is executed by an operating system (not shown) for each incoming packet 2, classified either as real-time (RT) or best-effort (BE), from the packet classifier 310 at step 400. When an IP packet is input from the packet classifier 310, the packet discarding block 320 determines whether the RT and BE queues 330A-330B are empty ($len_{rt} + len_{be} = 0$; and $t\text{-}t_{free} > t_{free}$) at step 410. If the RT and BE queues 330A-330B are not empty at step 410, the packet discarding block 320 proceeds to step 414. If the RT and BE queues 330A-330B are empty ($len_{rt} + len_{be} = 0$; and $t\text{-}t_{free} > t_{free}$) at step 410, however, the packet discarding block 320 reset counters p1 and p2 (p1 = p2 = 0) and the scheduled transmission time of the next packet ($t_{free} := t\text{-}t_{free}$) at step 412.

Next, the packet discarding block 320 determines whether the RT latency is larger than the maximum latency value for RT queue 330A ($latency(len_{rt}) > LAT_{rt}$) at step 414. If the RT latency is too large, the packet discarding block 320 discards the input RT packet at step 416. If the RT latency is not too large at step 414, the packet discarding block 320 proceeds to determine whether the total latency is larger than the maximum latency value for the complete queue system 330 ($latency(len_{rt} + len_{be}) > LAT_{tot}$) at step 418. If the total latency is too large, the packet discarding block 320 adds a BE packet to the tail of BE queue 330B (add_to_BE(p); and p :=head_BE()) at step 420 and discards the packet from the BE queue head at step 416.

If the total latency is not too large at step 418, the packet discarding block 320 determines whether an input packet 2 is a RT packet and whether the RT queue length is longer than the maximum queue length (tp=RT; and len_{rt}>$QLEN_{rt}$) at step 422. If the input packet is a RT packet and the RT queue length is too large, the packet discarding block 320 discards the input RT packet at step 416.

Likewise, if the RT queue length is not too large or tp = BE at step 422, the packet discarding block 320 determines whether an input packet is a BE packet and whether the BE queue length is larger than the maximum queue length (tp=BE; and len_{be}>$QLEN_{be}$) at step 424. If the input packet is a BE packet and the BE queue length is too large (tp=BE; and len_{be}>$QLEN_{be}$), the packet discarding block 320 adds the input packet to the tail of BE queue 330B (add_to_BE(p); and p := head_BE()) at steps 426 and 428 and discards the packet from the BE queue head at step 430.

If the BE queue length is not too large or tp = RT at block 424, the packet discarding block 320 proceeds to determine whether an input packet is a RT packet and whether there is a concurrency situation (tp = RT; and lp <p1) at step 432. If there is a concurrency situation, the packet discarding block 320 checks p1 counter (p1 := p1 - l_p) at step 434 and discards a RT packet if the p1 counter indicates that it is time to discard the RT packet at step 430.

If there is no concurrency at step 432, the packet discarding block 320 proceeds to determine whether an input packet 2 is a RT packet or a BE packet at step 436. If the input packet 2 is a RT packet, the packet discarding block 320 places the input RT packet in the appropriate RT queue 330A (add_to_RT(p)) at step 438; otherwise, the BE queue 330B (add_to_BE(p)) at step 440.

The packet scheduling block 340 executes on a regular basis, for example, every 10 ms for sending packets to the outgoing device in accordance with the bandwidth (BW) parameters. The following scheduling algorithm describes the functionality of the packet scheduling block 340:

```
Release concurrency parameters if appropriate
if $len_{rt} + len_{be} = 0$
   then
         $p_1 = p_2 = 0$
fi;
Is it time to send a BE packet through during concurrency
$p = first\_of\_best\_effort()$;
if $(len_{be} \geq 0)$
$\wedge(I_p \leq p_2)$
   then
         if
                  $bits(BW, t - t_{free}) > l_p$
                  then
                           $t_{free} = t_{free} + time(BW, l_p)$;
                           send $(p)$;
                  else
                           continue;
                  fi
fi
Send RT packet
$p = first\_of\_real\_time()$;
if $(len_{rt} \geq 0)$
   then
         if
                  $bits(BW, t - t_{free} > l_p$
                  then
                           $t_{free} = t_{free} + time(BW, l_p)$;
                           send $(p)$;
                  else
                           continue;
                  fi
fi
Send BE packet
$p = first\_of\_best\_effort()$;
if$(len_{be} \geq 0)$
   $\wedge bits(BW, t - t_{free}) > l_p$
   then
         $t_{free} = t_{free} + time(BW, l_p)$;
         send$(p)$;
fi;
```

Basically, the IP packet scheduling algorithm as described provides several underlying functions: The first function relates to determining if both the real-time (RT) and best-effort (BE) queues 330A-330B are empty and counters p1 and p2 are zero. The second function relates to the BE packet and determining whether there is concurrency (both RT and BE queues 330A-330B are active) and p2 counter indicates that a BE packet is to be transmitted. If there is concurrency and p2 counter indicates that a BE packet can be transmitted, checking if total bandwidth constraint allows sending of the BE packet from the BE queue head. If transmission is allowed, updating the time of next transmission and transmitting the BE packet. The third function relates to the RT packet and determining whether the RT queue 330A is not empty and transmission timer is permitted. If the RT queue 330A is not empty and transmission timer allows, updating the transmission timer and sending the RT packet from the RT queue head. The fourth function relates to the BE packet and determining whether the BE queue 330B is not empty and transmission timer is permitted. If the BE queue 330B is not empty and transmission timer allows, updating the transmission timer and sending the BE packet from the BE queue head.

Referring to FIG. 5, a detailed flowchart representation of an IP packet scheduling algorithm of the packet scheduling block 340 is illustrated. The algorithm of the packet scheduling block 340 is executed periodically by an operating system (not shown) to transmit IP packets from the BE and RT queues 330A-330B. The packet scheduling block 340 determines whether both BE and RT queues 330A-330B are empty (len_{rt} + len_{be} = 0) at step 500 and whether p1 and p2 counters are set at zero (p1 = p2 = 0) at step 502.

After p1 and p2 counters are set at zero (p1 = p2 = 0) at step 502, the packet scheduling block 340 determines proceeds to select the BE queue head (p = head_BE()) at step 504. Next, the packet scheduling block 340 determines if p2 counter indicates that a BE packet can be transmitted ($len_{be} > 0$; and $l_p \leq p2$) at step 506. If p2 counter indicates that a BE packet can be transmitted, the packet scheduling block 340 checks if the total bandwidth constraint allows sending of the BE packet from the BE queue head ($bits(BW,t - t_{free}) > l_p$) at step 508. If BE packet transmission is allowed, the packet scheduling block 340 updates the time of next transmission ($t_{free} = t_{free}$ + time (BW, l_p)) at step 510, and sends the BE packet at step 512.

Next, the packet scheduling block 340 selects the RT queue head (p = head_RT()) at step 514, and determines if the RT queue 330A is not empty at step 516. If the RT queue 330A is not empty, the packet scheduling block 340 determines if the total bandwidth constraint allows sending of the RT packet from the RT queue head ($bits(BW,t - t_{free}) > l_p$) at step 518. If RT packet transmission is allowed, the packet scheduling block 340 updates the time of next transmission ($t_{free} = t_{free}$ + time (BW, l_p)) at step 520, and sends the RT packet at step 522.

\

Next, the packet scheduling block 340 selects the BE queue head (p = head_BE()) at step 524, and determines if the total bandwidth constraint allows sending of the BE packet from the BE queue head (bits(BW,t - t_{free}) > l_p) at step 526. If BE packet transmission is allowed, the packet scheduling block 340 updates the time of next transmission ($t_{free} = t_{free}$ + time (BW, l_p)) at step 528, and sends the BE packet at step 530.

The operation of the traffic shaping algorithm (including a packet discarding algorithm and a packet scheduling algorithm) as represented by the functional block diagram as shown in FIG. 3 has been verified by simulations using an operating system (OS) such as the Linux open-source OS. A 200-byte real-time packet every 20 ms and a 1500-byte BE packet every 100 ms were provided to the traffic shaping algorithm. The number of discarded, queued and scheduled packets of each type was recorded. An emulated jiffy length (shaper call frequency) of 10 ms was used in the simulations. The shaper output bandwidth was averaged over a period of 50 packets or 1000 ms.

The results can be seen in FIG. 6, which illustrates the shaper output bandwidth (bits/second) of the advanced traffic shaper as a function of time (second). The traffic shaper 300 is able to use the bandwidth BW efficiently. As shown in FIG. 6, the traffic shaper 300 implements the BE keep-alive bandwidth, while maintaining -- on the average -- the total bandwidth constraint. The following parameters are used: the total bandwidth, BW_{tot} = 80 kbits/s, the bandwidth for BE traffic which is maintained even at times when there is enough RT traffic to fill the BW of the traffic shaper 300, $BW_{BE,min}$ = 2 kbits/2; the maximum latency value for the RT queue 330A, LAT_{rt} = 50 ms, and the maximum latency value for the complete queue system 330, LAT_{tot} = 150 ms.

FIG. 7 illustrates the length of RT queue 330A as a function of packet sequence number. A BE packet is scheduled roughly every 500 packets, and at those times, RT packets must be queued up. For demonstration purposes, a higher latency value for the RT queue of LAT_{rt} = 100 ms is used. Other parameters remain the same as those described in FIG. 6. For example, BW_{tot} = 80 kbits/s, $BW_{BE,min}$ = 2 kbits/2, and LAT_{tot} = 150 ms.

FIG. 8 shows the cummulative number of discarded real-time packets as a function of packet sequence number for a case in which 20% of RT packets must be discarded and RT queue latency is low (50 ms). The parameters are used: BW_{tot} = 64 kbits/s, $BW_{BE,min}$ = 2 kbits/2, LAT_{rt} = 50 ms, and LAT_{tot} = 150 ms. The figure is centered at the moment when BE packet is scheduled and RT packets must be discarded. It can be seen that the algorithm does not discard two adjacent RT packets.

FIG. 9 illustrates one preferred usage implementation of an IP-based network of a source terminal and a destination terminal having an advanced traffic shaper incorporated therein for controlling traffic bandwidth usage to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention. As shown in FIG. 9, a source terminal 710 may be part of a private data network 700 such as an ISDN for sending IP packets to a destination terminal 900. An advanced traffic shaper 300 of the present invention may be installed at a network egress 720 of the data network 700 for traffic shaping the IP packets before sending to the destination terminal 900 via an Internet 820. The network egress 720 may also be a source terminal for sending IP packets to the destination terminal 900.

A bandwidth-limited link 810 may be used to provide connection between the data network 700 and the Internet 820. The IP bandwidth-limited link 10 may represent a low bandwidth link such as, for example, an integrated service digital service (ISDN) link and a modem link that connect to the network egress 720 of the data network 700 and the network ingress 822 of the Internet 820. A network ingress 822 may be an Internet Service Provider (ISP) Server provided to receive IP packets from the source terminal 710 of the data network 700. The network ingress 822 may include an advanced traffic shaper 300 of the present invention for traffic shaping of real-time traffic from the Internet side toward the source terminal 710.

A high-speed link 830 may be used to provide connection between the Internet 820 and the destination terminal 900. The high-speed link 830 may represent a high capacity Internet link of over 100 Mb/s that does not need to prioritize real-time traffic. Accordingly, the destination terminal 900 may not require the advanced traffic shaper 300 of the present invention.

FIG. 10 illustrates another preferred usage implementation of an IP-based network of a source terminal and a destination terminal having an advanced traffic shaper incorporated therein for controlling traffic bandwidth usage to guarantee proper QoS (Quality of Service) for different types of real-time traffic applications while concomitantly providing keep-alive bandwidth for best-effort traffic according to the principles of the present invention. As shown in FIG. 10, the source terminal 710 may be a stand-alone PC for home usage for sending IP packets to the destination terminal 900 via the Internet 820. The source terminal 710 may be connected to the Internet 820 through the bandwidth-limited link 810 and the network ingress 822. An advanced traffic shaper 300 of the present invention may be installed at the source terminal 710 for traffic shaping the IP packets before sending to the destination terminal 900 via an Internet 820. Similarly, the network ingress 822 may include an advanced traffic shaper 300 of the present invention for traffic shaping of real-time traffic from the Internet side toward the source terminal 710.

As described from the foregoing, the present invention advantageously provides a real-time traffic shaper configured for priority forwarding of real-time traffic while providing keep-alive bandwidth for "best-effort" traffic at the same time.

While there have been illustrated and described what are considered to be preferred embodiments of the present invention, it will be understood by those skilled in the art that various changes and modifications may be made, and equivalents may be substituted for elements thereof without departing from the true scope of the present invention. Further, many modifications may be made to adapt a particular situation to the teachings of the present invention without departing from the central scope of the present invention. Therefore, it is intended that the present invention not be limited to the particular embodiment disclosed as the best mode contemplated for carrying out the present invention, but that the present invention includes all embodiments falling within the scope of the appended claims.

CLAIMS

What is claimed is:

1. A traffic shaper for shaping traffic in a packet network, comprising:

a packet classifier coupled to receive an incoming packet from said packet network and to classify the incoming packet as one of a real-time packet and a best-effort packet;

a packet discarding block coupled to perform traffic shaping of the incoming packet from the packet classifier and discard the incoming packet in accordance with traffic shaping parameters;

connection queues coupled to temporary store real-time packets and best-effort packets from the packet discarding block, respectively; and

a packet scheduling block coupled to send the real-time and best-effort packets from the connection queues to an outgoing device in accordance with said traffic shaping parameters.

2. The traffic shaper as claimed in claim 1, wherein said connection queues includes a real-time queue for storing the real-time packets and a best-effort queue for storing the best-effort packets.

3. The traffic shaper as claimed in claim 2, wherein said packet discarding block includes packet counters used to balance real-time packet discards in situations where there is concurrency when both the real-time and best-effort queues are active.

4. The traffic shaper as claimed in claim 3, wherein said packet discarding block is executed to discard an incoming packet by:

determining whether the real-time and best-effort queues are empty;

when the real-time and best-effort queues are empty, resetting the counters and a scheduled transmission time of the next incoming packet;

determining whether a real-time latency is greater than a maximum latency value of the real-time queue;

when the real-time latency is greater than the maximum latency value of the real-time queue, discarding the real-time packet input from the packet classifier;

determining whether a total latency is greater than a maximum latency value of both the real-time and best-effort queues;

when the total latency is greater than the maximum latency value of both the real-time and best-effort queues, adding a best-effort packet to a tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue;

determining whether a length of the real-time queue is greater than a maximum queue length of the real-time queue;

when the length of the real-time queue is greater than a maximum queue length of the real-time queue, discarding the real-time packet input from the packet classifier;

determining whether a length of the best-effort queue is greater than a maximum queue length of the best-effort queue;

when the length of the best-effort queue is greater than a maximum queue length of the best-effort queue, adding a best-effort packet input from the packet classifier to the tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue;

determining whether there is concurrency and time to discard a real-time packet to make room for best-effort packets;

when there is concurrency and time to discard the real-time packet, discarding the real-time packet to implement the keep-alive bandwidth for best-effort packets; and placing the real-time packet that was not discarded in the real-time queue.

5. The traffic shaper as claimed in claim 3, wherein said packet scheduling block is periodically executed to schedule output of an incoming packet by:

determining whether both the real-time and best-effort queues are empty;

determining whether there is concurrency and time to transmit the best-effort packet from the best-effort queue;

when there is concurrency and time to transmit the best-effort packet, checking whether total bandwidth constraint allows transmission of the best-effort packet from the best-effort queue head;

when transmission is allowed, updating a time of next transmission and transmitting the best-effort packet to the outgoing device;

determining whether the real-time queue is not empty and time for transmission of the real-time packet is permitted;

when the real-time queue is not empty and time for transmission of the real-time packet is permitted, updating the transmission time and transmitting the real-time packet to the outgoing device from the real-time queue head;

determining whether the best-effort queue is not empty and time for transmission of the best-effort packet is permitted; and when the best-effort queue is not empty and time for transmission of the best-effort packet is permitted, updating the transmission time and transmitting the best-effort packet to the outgoing device from the best-effort queue head.

6. The traffic shaper as claimed in claim 1, wherein said connection queues correspond to packet buffers constructed from first-in-first-out (FIFO) memory for storing real-time packets and best-effort packets, respectively.

7. A packet network, comprising:

a source terminal comprising a host and a network interface controller;

a destination terminal comprising a host and a network interface controller;

a bandwidth-limited link providing connection between the source terminal and the destination terminal; and

at least one network interface controller of the source terminal or the destination terminal comprises a traffic shaper installed therein for shaping traffic in the packet network, said traffic shaper comprising:

a packet classifier coupled to receive an incoming packet and to classify the incoming packet as one of a real-time packet and a best-effort packet;

a packet discarding block coupled to perform traffic shaping of the incoming packet from the packet classifier and discard the incoming packet in accordance with traffic shaping parameters;

connection queues coupled to temporary store real-time packets and best-effort packets from the packet discarding block, respectively; and

a packet scheduling block coupled to send the real-time and best-effort packets from the connection queues to an outgoing device in accordance with said traffic shaping parameters.

8. The data network as claimed in claim 7, wherein said connection queues includes a real-time queue for storing the real-time packets and a best-effort queue for storing the best-effort packets.

9. The data network as claimed in claim 7, wherein said bandwidth-limited link corresponds to one of a digital services digital network (ISDN) link and a modem link with low-bandwidth for providing connection between the source terminal and the destination terminal.

placing the real-time packet that was not discarded in the real-time queue.

13. The data network as claimed in claim 8, wherein said packet scheduling block is periodically executed to schedule output of an incoming packet by:

determining whether both the real-time and best-effort queues are empty;

determining whether there is concurrency and time to transmit the best-effort packet from the best-effort queue;

when there is concurrency and time to transmit the best-effort packet, checking whether total bandwidth constraint allows transmission of the best-effort packet from the best-effort queue head;

when transmission is allowed, updating a time of next transmission and transmitting the best-effort packet to the outgoing device;

determining whether the real-time queue is not empty and time for transmission of the real-time packet is permitted;

when the real-time queue is not empty and time for transmission of the real-time packet is permitted, updating the transmission time and transmitting the real-time packet to the outgoing device from the real-time queue head;

determining whether the best-effort queue is not empty and time for transmission of the best-effort packet is permitted; and

when the best-effort queue is not empty and time for transmission of the best-effort packet is permitted, updating the transmission time and transmitting the best-effort packet to the outgoing device from the best-effort queue head.

14. The data network as claimed in claim 7, wherein said source terminal is connected to one of an integrated services digital network (ISDN).

15. The data network as claimed in claim 7, further comprising an Internet Service Provider (ISP) and an Internet disposed between the bandwidth-limited link and the destination terminal.

16. The data network as claimed in claim 7, wherein said traffic shaper is an algorithm configured for traffic shaping and avoiding correlated loss for real-time streams while providing keep-alive bandwidth for best-effort traffic.

17. A computer usable medium having computer readable program code means embodied therein a traffic shaper algorithm for use in a source terminal for shaping traffic in a packet network, comprising:

a packet classifier module configured to receive an incoming packet and to classify the incoming packet as one of a real-time packet and a best-effort packet;

a packet discarding module configured to perform traffic shaping of the incoming packet from the packet classifier and discard the incoming packet in accordance with traffic shaping parameters;

a connection queue module configured to temporary store real-time packets and best-effort packets from the packet discarding block, respectively; and

a packet scheduling module configured to send the real-time and best-effort packets from the connection queues to an outgoing device in accordance with said traffic shaping parameters.

18. The computer usable medium as claimed in claim 17, wherein said connection queue module contain a real-time queue for storing the real-time packets and a best-effort queue for storing the best-effort packets.

19. The computer usable medium as claimed in claim 17, wherein said packet discarding module is configured to discard an incoming packet by:

determining whether the real-time and best-effort queues are empty;

when the real-time and best-effort queues are empty, resetting the counters and a scheduled transmission time of the next incoming packet;

determining whether a real-time latency is greater than a maximum latency value of the real-time queue;

when the real-time latency is greater than the maximum latency value of the real-time queue, discarding the real-time packet input from the packet classifier;

determining whether a total latency is greater than a maximum latency value of both the real-time and best-effort queues;

when the total latency is greater than the maximum latency value of both the real-time and best-effort queues, adding a best-effort packet to a tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue;

determining whether a length of the real-time queue is greater than a maximum queue length of the real-time queue;

when the length of the real-time queue is greater than a maximum queue length of the real-time queue, discarding the real-time packet input from the packet classifier;

determining whether a length of the best-effort queue is greater than a maximum queue length of the best-effort queue;

10. The data network as claimed in claim 7, wherein said connection queues correspond to packet buffers constructed from first-in-first-out (FIFO) memory for storing real-time packets and best-effort packets, respectively.

11. The data network as claimed in claim 8, wherein said packet discarding block includes packet counters used to balance real-time packet discards in situations where there is concurrency when both the real-time and best-effort queues are active.

12. The data network as claimed in claim 8, wherein said packet discarding block is executed to discard an incoming packet by:

determining whether the real-time and best-effort queues are empty;

when the real-time and best-effort queues are empty, resetting the counters and a scheduled transmission time of the next incoming packet;

determining whether a real-time latency is greater than a maximum latency value of the real-time queue;

when the real-time latency is greater than the maximum latency value of the real-time queue, discarding the real-time packet input from the packet classifier;

determining whether a total latency is greater than a maximum latency value of both the real-time and best-effort queues;

when the total latency is greater than the maximum latency value of both the real-time and best-effort queues, adding a best-effort packet to a tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue;

determining whether a length of the real-time queue is greater than a maximum queue length of the real-time queue;

when the length of the real-time queue is greater than a maximum queue length of the real-time queue, discarding the real-time packet input from the packet classifier;

determining whether a length of the best-effort queue is greater than a maximum queue length of the best-effort queue;

when the length of the best-effort queue is greater than a maximum queue length of the best-effort queue, adding a best-effort packet input from the packet classifier to the tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue;

determining whether there is concurrency and time to discard a real-time packet to make room for best-effort packets;

when there is concurrency and time to discard the real-time packet, discarding the real-time packet to implement the keep-alive bandwidth for best-effort packets; and when the length of the best-effort queue is greater than a maximum queue length of the best-effort queue, adding a best-effort packet input from the packet classifier to the tail of the best-effort queue and discarding a best-effort packet from a head of the best-effort queue;

determining whether there is concurrency and time to discard a real-time packet to make room for best-effort packets;

when there is concurrency and time to discard the real-time packet, discarding the real-time packet to implement the keep-alive bandwidth for best-effort packets; and

placing the real-time packet that was not discarded in the real-time queue.

20. The computer usable medium as claimed in claim 17, wherein said packet scheduling block is configured to periodically schedule output of an incoming packet by:

determining whether both the real-time and best-effort queues are empty;

determining whether there is concurrency and time to transmit the best-effort packet from the best-effort queue;

when there is concurrency and time to transmit the best-effort packet, checking whether total bandwidth constraint allows transmission of the best-effort packet from the best-effort queue head;

when transmission is allowed, updating a time of next transmission and transmitting the best-effort packet to the outgoing device;

determining whether the real-time queue is not empty and time for transmission of the real-time packet is permitted;

when the real-time queue is not empty and time for transmission of the real-time packet is permitted, updating the transmission time and transmitting the real-time packet to the outgoing device from the real-time queue head;

determining whether the best-effort queue is not empty and time for transmission of the best-effort packet is permitted; and

when the best-effort queue is not empty and time for transmission of the best-effort packet is permitted, updating the transmission time and transmitting the best-effort packet to the outgoing device from the best-effort queue head.

FIG. 1

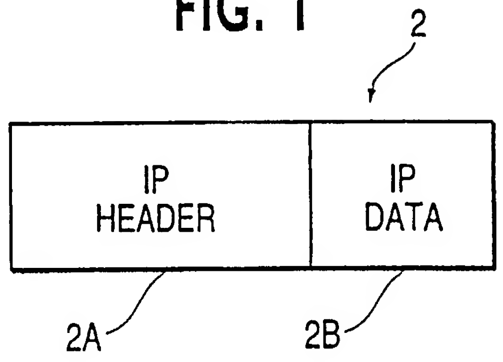

2
IP
HEADER
IP
DATA
2A
2B

FIG. 2

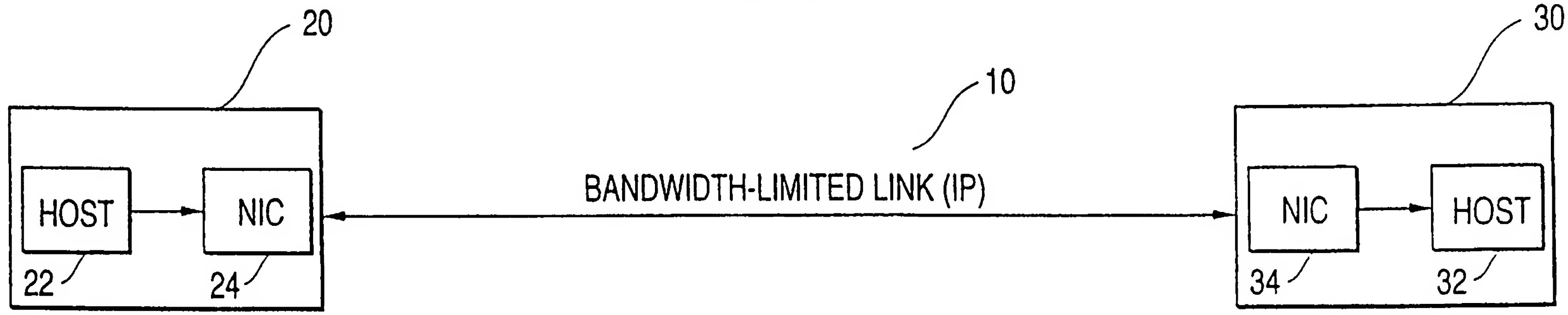

20
10
30
HOST
NIC
BANDWIDTH-LIMITED LINK (IP)
NIC
HOST
22
24
34
32

FIG. 3

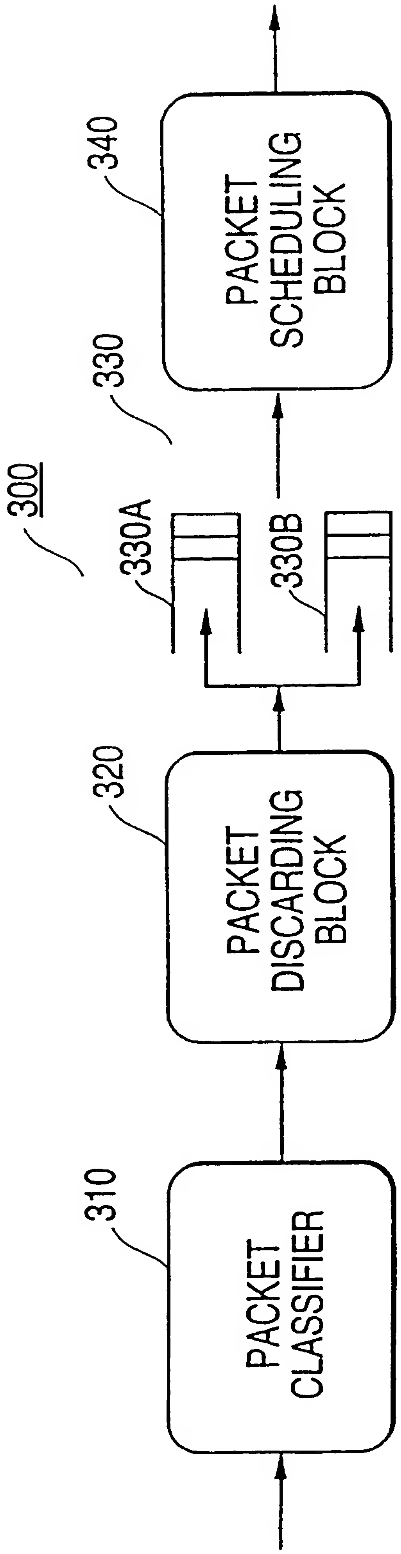

300
310
PACKET CLASSIFIER
320
PACKET DISCARDING BLOCK
330
330A
330B
340
PACKET SCHEDULING BLOCK

FIG. 4

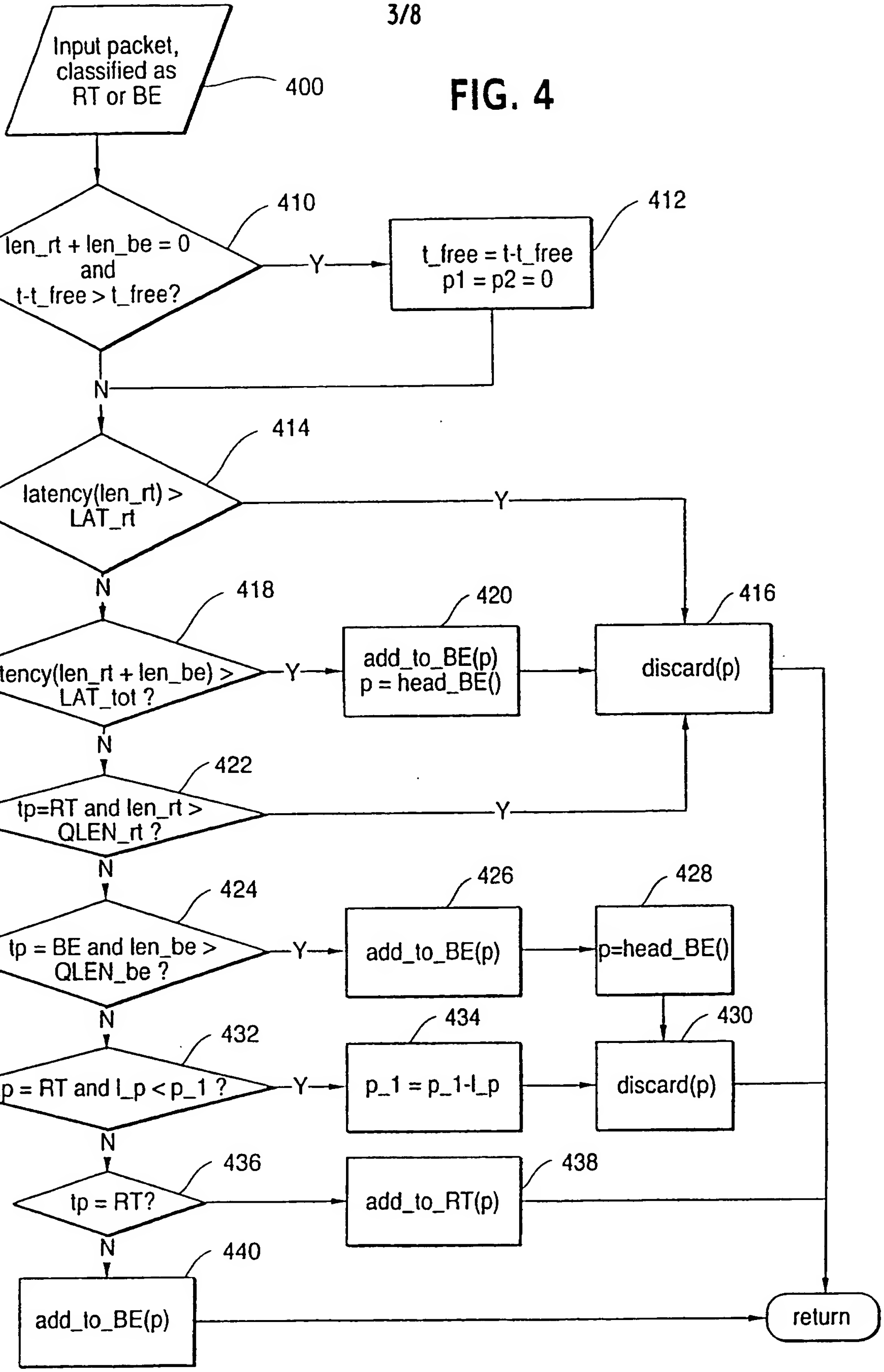

Input packet, classified as RT or BE
400
len_rt + len_be = 0 and t-t_free > t_free?
410
Y
t_free = t-t_free p1 = p2 = 0
412
N
latency(len_rt) > LAT_rt
414
Y
N
latency(len_rt + len_be) > LAT_tot ?
418
Y
add_to_BE(p) p = head_BE()
420
discard(p)
416
N
tp=RT and len_rt > QLEN_rt ?
422
Y
N
tp = BE and len_be > QLEN_be ?
424
Y
add_to_BE(p)
426
p=head_BE()
428
N
tp = RT and l_p < p_1 ?
432
Y
p_1 = p_1-l_p
434
discard(p)
430
N
tp = RT?
436
add_to_RT(p)
438
N
add_to_BE(p)
440
return

FIG. 5

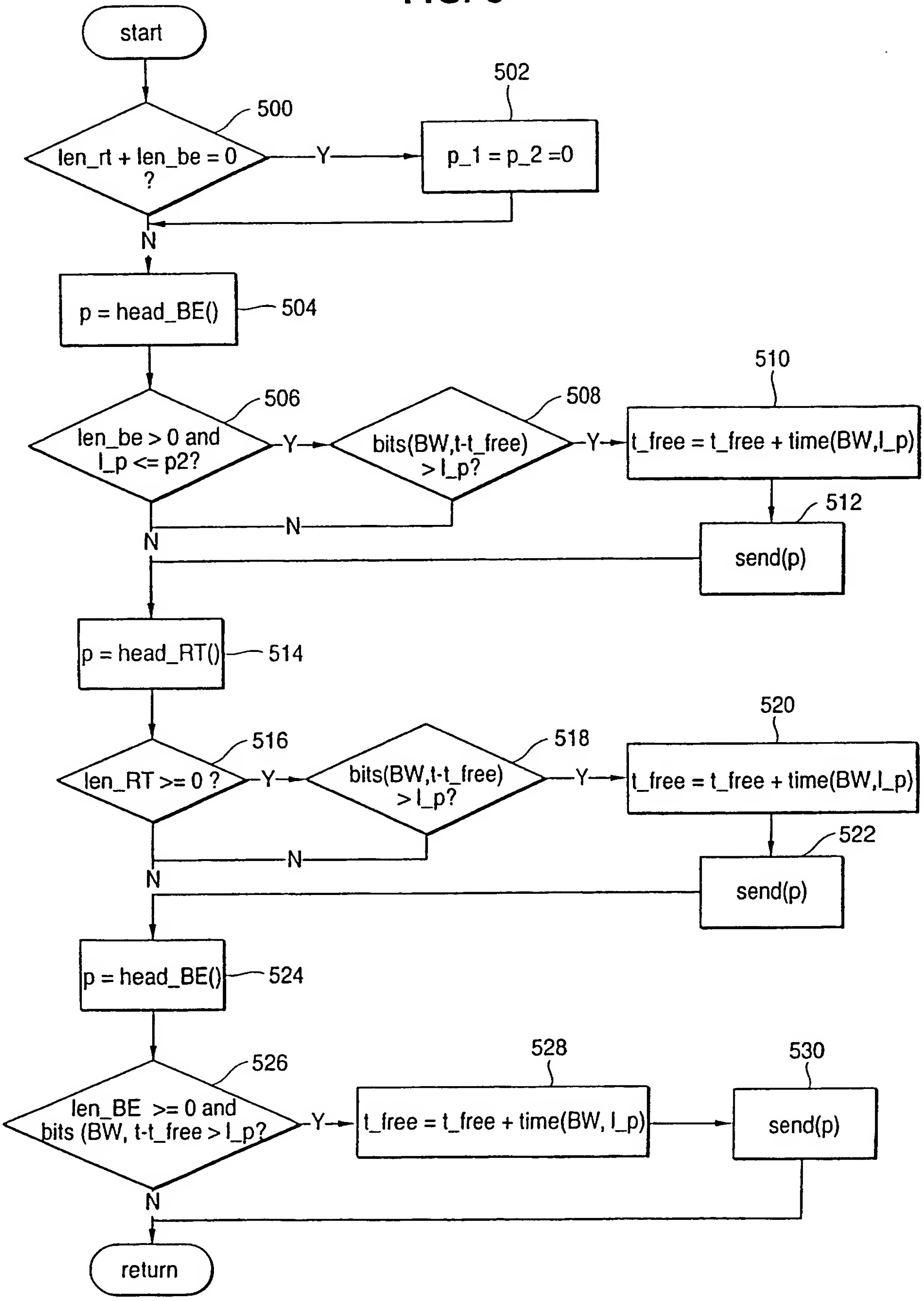

start
500
len_rt + len_be = 0 ?
Y
502
p_1 = p_2 =0
N
p = head_BE()
504
506
len_be > 0 and l_p <= p2?
Y
508
bits(BW,t-t_free) > l_p?
Y
510
t_free = t_free + time(BW,l_p)
512
send(p)
N
N
p = head_RT()
514
516
len_RT >= 0 ?
Y
518
bits(BW,t-t_free) > l_p?
Y
520
t_free = t_free + time(BW,l_p)
522
send(p)
N
N
p = head_BE()
524
526
len_BE >= 0 and bits (BW, t-t_free > l_p?
Y
528
t_free = t_free + time(BW, l_p)
530
send(p)
N
return

FIG. 6

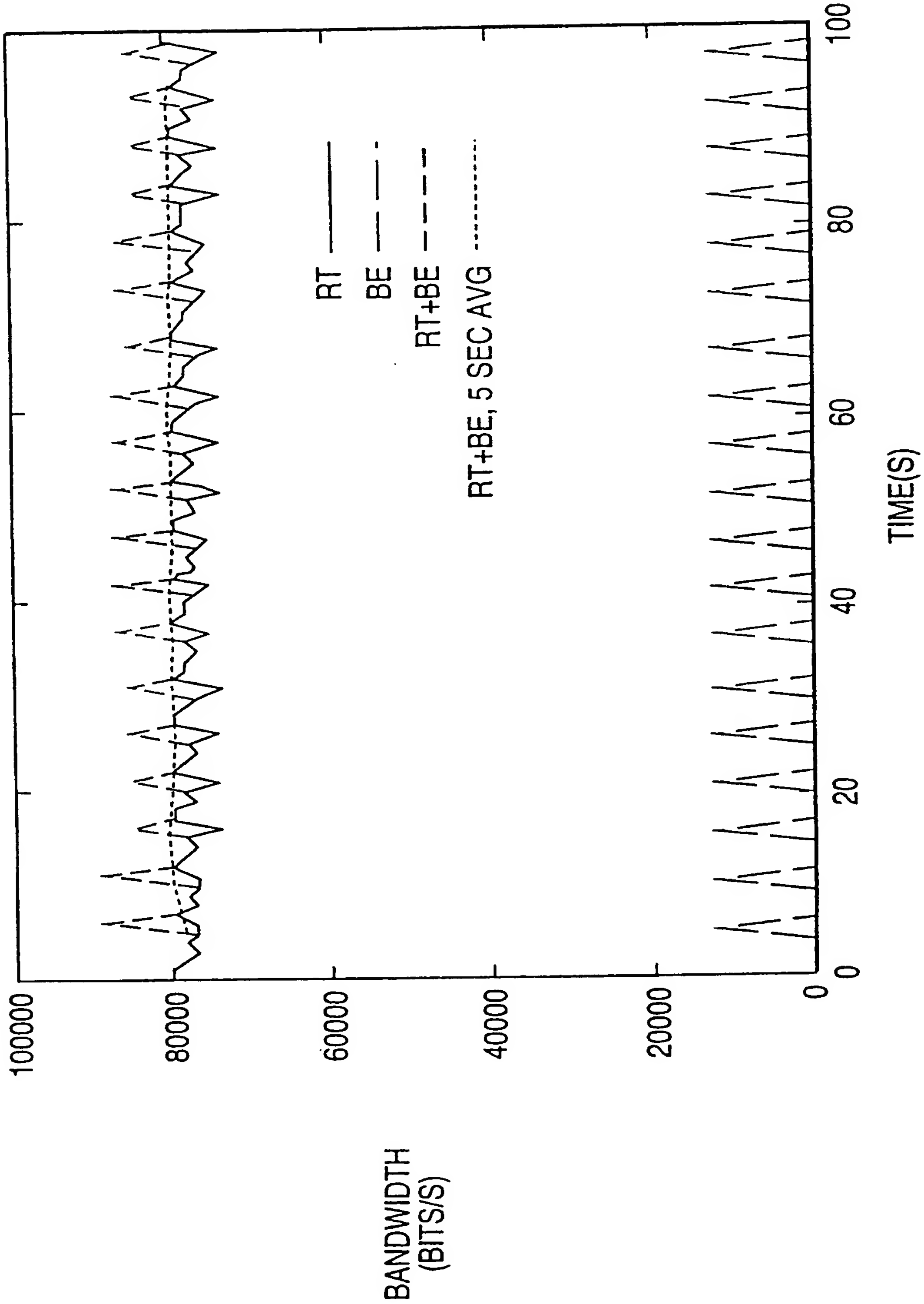

100000
80000
60000
40000
20000
0
BANDWIDTH (BITS/S)
0
20
40
60
80
100
TIME(S)
RT
BE
RT+BE
RT+BE, 5 SEC AVG

FIG. 7

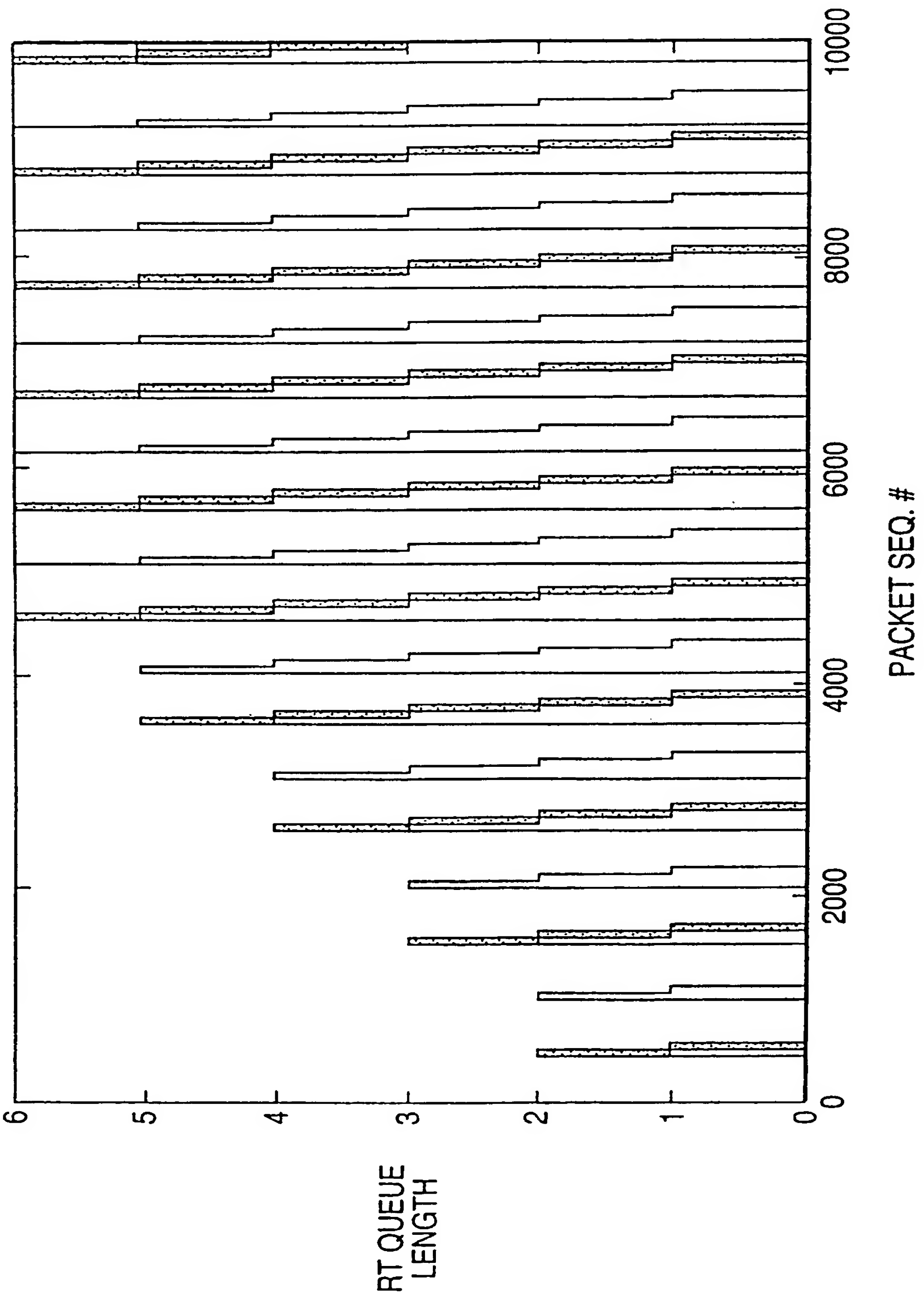
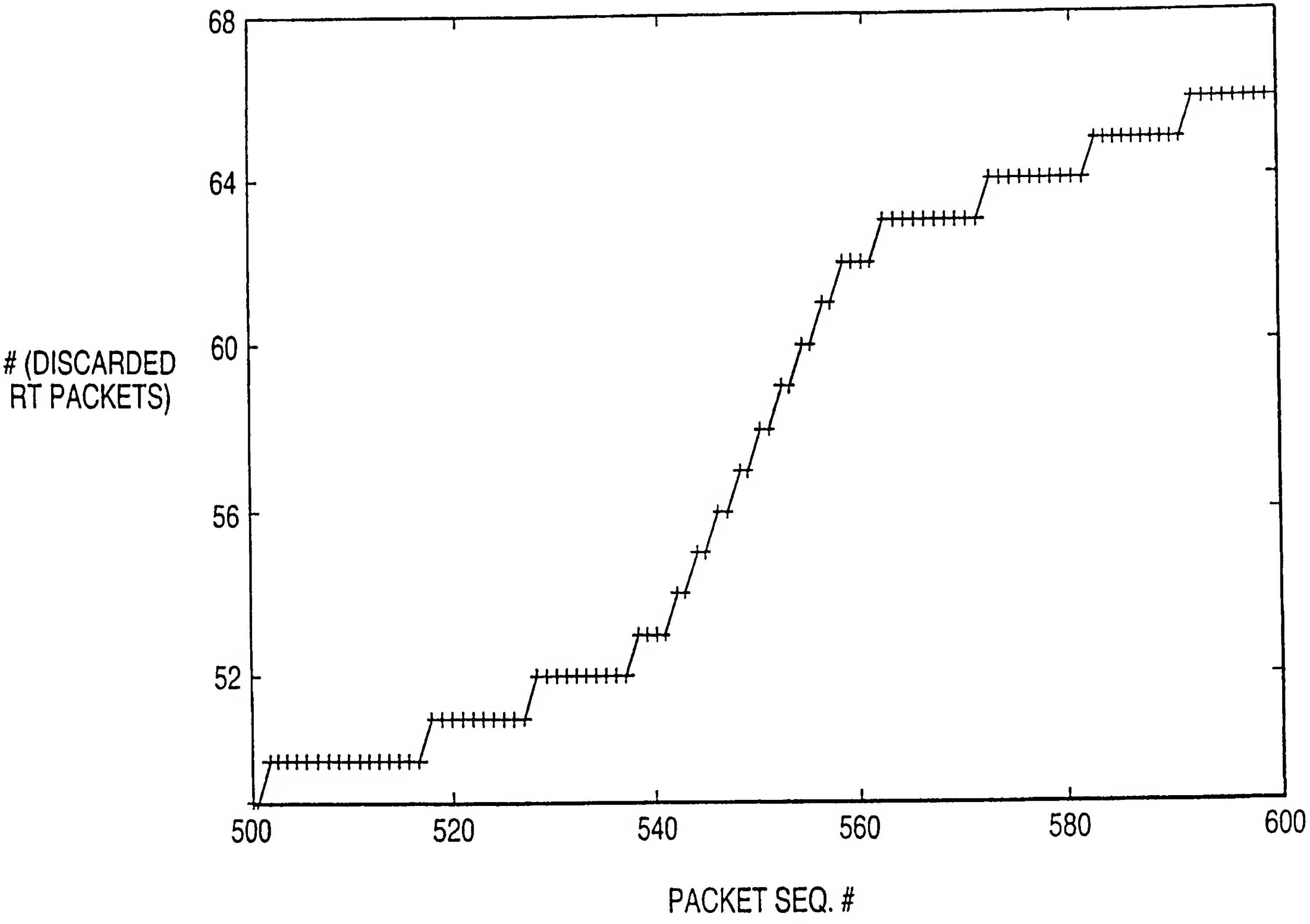

RT QUEUE LENGTH
0
1
2
3
4
5
6
PACKET SEQ. #
0
2000
4000
6000
8000
10000

FIG. 8


68
64
60
56
52
# (DISCARDED RT PACKETS)
500
520
540
560
580
600
PACKET SEQ. #

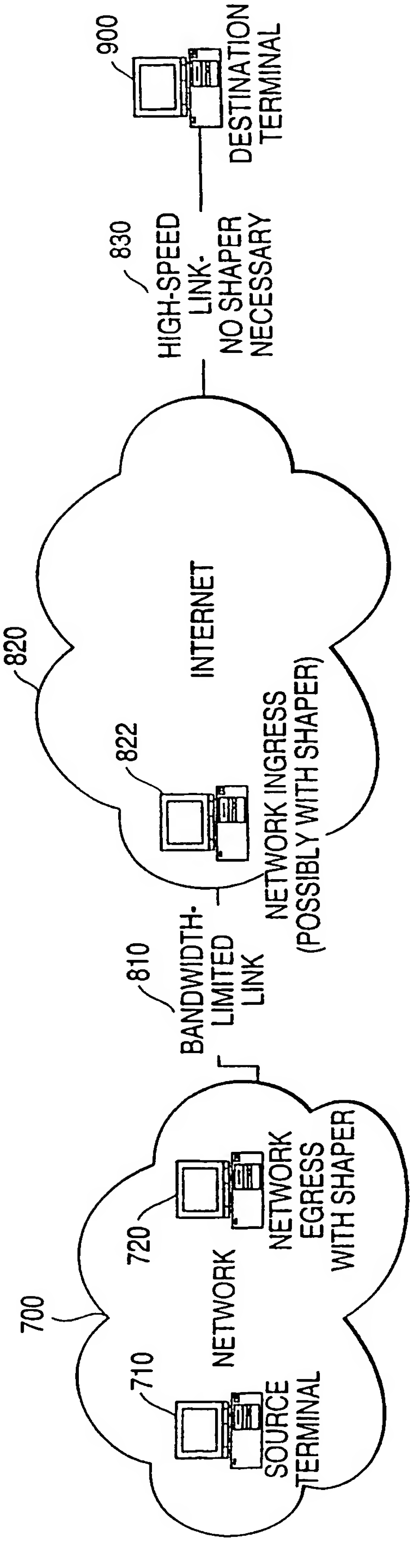

700
NETWORK
710
SOURCE TERMINAL
720
NETWORK EGRESS WITH SHAPER
810
BANDWIDTH-LIMITED LINK
820
822
NETWORK INGRESS (POSSIBLY WITH SHAPER)
INTERNET
830
HIGH-SPEED LINK-NO SHAPER NECESSARY
900
DESTINATION TERMINAL

FIG. 9

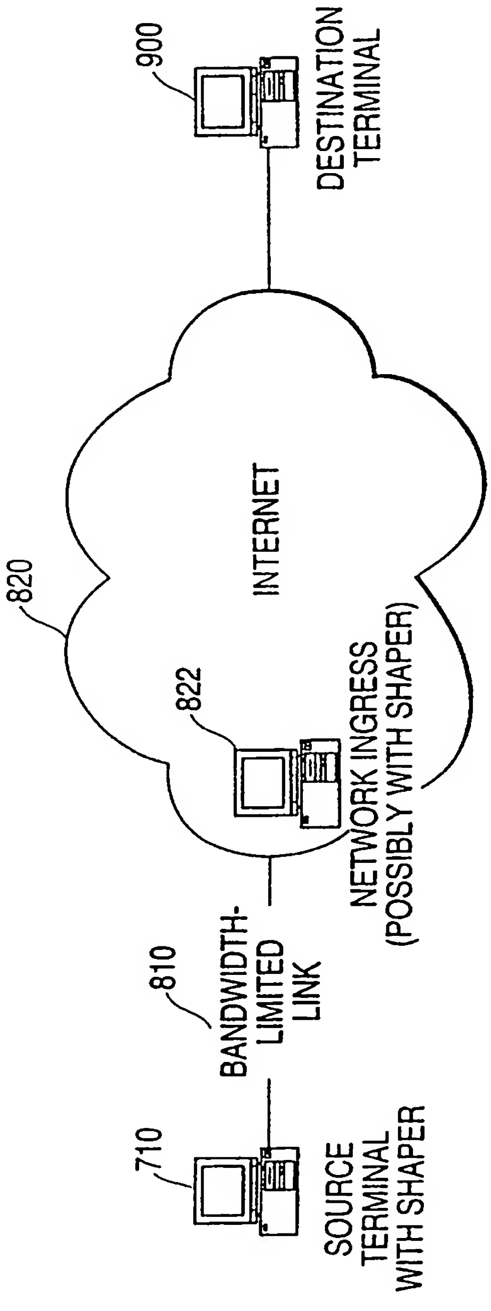

710
SOURCE TERMINAL WITH SHAPER
810
BANDWIDTH-LIMITED LINK
820
822
NETWORK INGRESS (POSSIBLY WITH SHAPER)
INTERNET
900
DESTINATION TERMINAL

FIG. 10

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 January 2001 (11.01.2001)

PCT

(10) International Publication Number
WO 01/03400 A3

(51) International Patent Classification⁷: H04L 12/56, 29/06

(21) International Application Number: PCT/IB00/00888

(22) International Filing Date: 1 July 2000 (01.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/346,747 2 July 1999 (02.07.1999) US

(71) Applicant *(for all designated States except US)*: NOKIA INTERNET COMMUNICATIONS INC. [US/US]; Robert C. Rolnik, 313 Fairchild Drive, Mountain View, CA 94043 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: RAISANEN, Vilho, I. [FI/FI]; Laajavudrenrinne 6 E 29, FIN-01620 Vantaa (FI). EKBERG, Jan-Erik [FI/FI]; Seljatie 1 A 5, FIN-00320 Helsinki (FI).

(74) Agent: ROLNIK, Robert; Nokia Inc., 6000 Connection Drive, 1-4-755, Irving, TX 75039 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

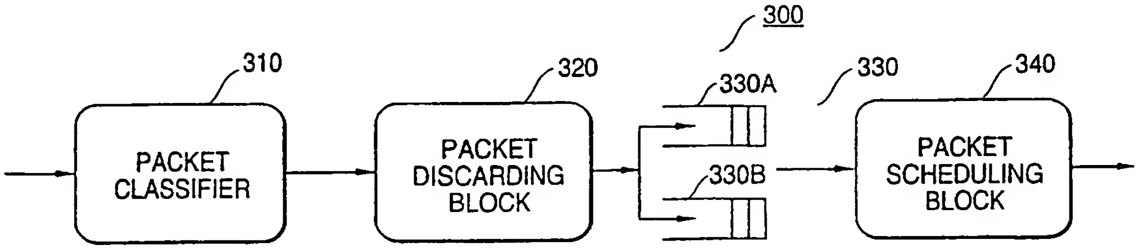
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report: 9 August 2001

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: REAL-TIME TRAFFIC SHAPER WITH GARANTEED BANDWIDTH FOR BEST-EFFORT TRAFFIC



(57) Abstract: An advanced traffic shaper is provided for shaping real-time traffic in an IP-based network while simultaneously providing keep-alive bandwidth for best-effort traffic. The traffic shaper comprises a packet classifier coupled to receive an incoming packet from the network and to classify the incoming packet as one of a real-time packet and a best-effort packet; a packet discarding block coupled to perform traffic shaping of the incoming packet from the packet classifier and discard the incoming packet in accordance with traffic shaping parameters; connection queues coupled to temporary store real-time packets and best-effort packets from the packet discarding block, respectively; and a packet scheduling block coupled to send the real-time and best-effort packets from the connection queues to an outgoing device in accordance with the traffic shaping parameters.

WO 01/03400 A3

INTERNATIONAL SEARCH REPORT

Inti onal Application No
PCT/IB 00/00888

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/56 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 28938 A (NORTHERN TELECOM LTD) 2 July 1998 (1998-07-02)	1,2,6-8, 10,17,18
A	page 4, line 17 -page 5, line 15	3,9, 11-16, 19,20

X	WO 99 13624 A (HODGKINSON TERENCE GEOFFREY ;BRITISH TELECOMM (GB); CARTER SIMON F) 18 March 1999 (1999-03-18)	1,2,6-8, 10,17,18
A	page 7, line 18 -page 10, line 14	3,9, 11-16, 19,20

	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 9 March 2001	Date of mailing of the international search report 16/03/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer RAMIREZ DE AREL.., F

INTERNATIONAL SEARCH REPORT

Int onal Application No
PCT/IB 00/00888

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
P,X	WO 00 30307 A (NORTEL NETWORKS CORP ;NICHOLS KATHLEEN M (US)) 25 May 2000 (2000-05-25)	1,2,6-8, 10,17,18
P,A	page 6, line 20 -page 11, line 2	3,9, 11-16, 19,20

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int onal Application No
PCT/IB 00/00888

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
WO 9828938	A	02-07-1998	US	6028842	A	22-02-2000
			WO	9828939	A	02-07-1998
			EP	0954943	A	10-11-1999
			JP	2000508145	T	27-06-2000
			US	6023456	A	08-02-2000
WO 9913624	A	18-03-1999	AU	9084398	A	29-03-1999
			EP	1013049	A	28-06-2000
WO 0030307	A	25-05-2000	AU	1525600	A	05-06-2000